AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) In a communication network, a method of TCP state migration comprising the steps of:

a) establishing a TCP/IP communication session between a client computer and a first server computer, said first server computer part of a plurality of server computers forming a web cluster containing information, said communication session established for the transfer of data contained within said information;

b) handing off said communication session to a selected server computer from said first server computer over a persistent control channel using TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer, ~~that~~ wherein the TCP handoff modules implement a TCP handoff protocol that works within kernel levels of ~~an existing TCP/IP protocol~~the operating systems; and

c) migrating a first TCP state of said first server computer to said selected server computer, and a second TCP state of said selected server computer to said first server computer over said control channel.

2. (Currently Amended) The method as described in Claim 1, wherein said step a) comprises the steps of:

receiving a SYN packet from said client computer at a first ~~BTCP~~ bottom TCP (BTCP) module located at said first server computer;

sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer;

receiving a first SYN/ACK packet from said first TCP module;

parsing said first ~~initial~~ TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with said TCP/IP communication session;

sending said SYN/ACK packet to said client computer;

receiving an ACK packet from said client at said first BTCP module;

sending said ACK packet to said first TCP module;

receiving a web request packet associated with said TCP/IP communication session at said first BTCP module at said first server computer;

storing said SYN packet, said ACK packet and said web request packet at said first server computer.

3. (Currently Amended) The method as described in Claim 2, wherein said step b) comprises the steps of:

examining content of said web request packet;

determining which of said plurality of server computers~~, a selected server computer,~~ can best process said ~~WEB~~ web request packet, based on said content, wherein the server computer determined to be able to best process said web request packet is the selected server computer;

sending a handoff request packet from said first BTCP module to a second BTCP module at said selected server computer over said control channel, if said selected server computer is not said first server computer;

including said SYN packet and said ACK packet in said handoff request packet;

changing a ~~first~~ destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module;

sending said SYN packet from said second BTCP module to ~~said~~ a second TCP module at said selected server computer;

receiving a second SYN/ACK packet at said second BTCP module;

parsing said second ~~initial~~ TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session;

changing a ~~second~~ destination IP address of said ACK packet to said second IP address, at said second BTCP module;

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session;

sending said ACK packet that is updated from said second BTCP module to said second TCP module; and

sending a handoff acknowledgment message from said second BTCP module to said first BTCP module.

4. (Currently Amended) The method as described in Claim 3, wherein step c) comprises ~~the steps of~~:

~~monitoring traffic associated with establishing said TCP/IP communication session in step a), at said first BTCP module, to parse a first initial TCP state of said first server computer, said first initial TCP state associated with said TCP/IP communication session; and~~

migrating said first ~~initial~~ TCP state to said second BTCP module over said control channel by including said first initial ~~TCP state~~ sequence number in said handoff request packet, ~~said first initial TCP state including a first sequence number,~~ such that said second BTCP module can calculate said first TCP state for said first server computer in said TCP/IP communication session.

5. (Currently Amended) The method as described in Claim 3, wherein step c) comprises ~~the steps of~~:

~~monitoring traffic associated with handing off said TCP/IP communication session at said second BTCP module, to parse a second initial TCP state of said selected server computer, said second initial TCP state associated with said TCP/IP communication session; and~~

migrating said second ~~initial~~ TCP state of said selected server computer to said first BTCP module by including said second initial ~~TCP state~~ sequence number in said handoff acknowledgment ~~packet~~message, ~~said second initial TCP state including a second initial sequence number,~~ such that said first BTCP module can calculate said second TCP state for said selected server computer in said TCP/IP communication session.

6. (Currently Amended) The method as described in Claim 2, comprising the further steps of:

intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module at the first server computer, said connection indication message sent by said first TCP module upon establishing said communication session; and

holding said connection indication message at said first UTCP module, wherein said first UTCP module and said first BTCP module provide a wrapper around said first TCP module.

7. (Currently Amended) The method as described in Claim 6, wherein said method comprises the further steps of:

sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment ~~packet~~ message to said first TCP module;

discarding said connection indication message at said first UTCP module;

receiving incoming data packets from said client at said first BTCP module;

changing ~~said~~ destination addresses of said incoming data packets to said second IP address;

updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer; and

forwarding said updated data packets to said selected server computer.

8. (Currently Amended) The method as described in Claim 6, comprising the further steps of:

sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer;

sending incoming data packets, including said web request packet, from said client computer, received at said first BTCP module, upstream.

9. (Currently Amended) The method as described in Claim [[1]]2, comprising the further ~~step~~steps of:

intercepting outgoing response packets from said selected server computer at a second bottom TCP (BTCP) module located at said selected server computer;

changing, by the second BTCP module, source addresses of said response packets to a first IP address of said first server computer;

updating, by the second BTCP module, sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer; and

sending said updated response packets to said client computer without passing the updated response packets through the first server computer.

10. (Currently Amended) The method as described in Claim [[1]]9, comprising the further steps of:

monitoring TCP/IP control traffic for said communication session at said second BTCP module;

understanding when said communication session is closed at said ~~second~~ selected server computer;

sending a termination message to said first server computer over said control channel;

terminating said TCP/IP communication session at said first server computer by terminating a forwarding mode at said first BTCP module; and

freeing data resources associated with said communication session at said first server computer.

11. (Currently Amended) In a communication network, a method of TCP state migration comprising the steps of:

a) establishing a TCP/IP communication session between a client computer and a first server computer, said first server computer part of a plurality of server computers forming a web cluster containing information, said communication session established for the transfer of data contained within said information;

b) monitoring traffic associated with establishing said TCP/IP communication session to understand a first initial TCP state of said first server computer associated with said TCP/IP communication session, at a first bottom-TCP (BTCP) module at said first server computer, wherein said first BTCP module is a dynamically loadable kernel module (DLKM) loaded in said first server computer without modifying a first operating system of said first server computer;

c) receiving a web request associated with said TCP/IP communication session at said first BTCP module at said first server computer;

d) examining content of said web request;

e) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content;

f) handing off said communication session to said selected server computer from said first server computer over a persistent control channel, if said selected server computer is not said first server computer;

g) monitoring traffic associated with handing off said TCP/IP communication session to understand a second initial TCP state of said selected server computer associated with said TCP/IP communication session, at a second BTCP module at said selected server computer, wherein said second BTCP module is a DLKM loaded in said selected server computer without modifying a second operating system of said selected server computer;

h) migrating said first initial TCP state to said selected server computer over said control channel, such that said second BTCP module can calculate a first TCP state for said first server computer in said TCP/IP communication session;

i) sending a second initial TCP state of said selected server computer to said first BTCP module, such that said first BTCP module can calculate a second TCP state for said selected server computer in said TCP/IP communication session;

j) forwarding data packets received at said first BTCP module from said client to said selected server computer, by changing said data packets to reflect said second TCP state and a second IP address of said selected server computer;

k) sending response packets from said selected server computer directly to said client computer by changing said response packets to reflect said first TCP state and a first IP address of said first server computer; and

l) terminating said TCP/IP communication session at said first server computer when said TCP/IP communication session is closed.

12. (Currently Amended) The method as described in Claim 11, wherein said ~~step~~steps a) and b) comprise~~comprises~~ the steps of:

receiving a SYN packet from said client computer at said first BTCP module;

sending said SYN packet upstream to a first TCP module located above said first BTCP module in [[a]]said first operating system of said first server computer;

receiving a first SYN/ACK packet from said first TCP module;

parsing, by the first BTCP module, said first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with[[,]] said TCP/IP communication session;

sending said SYN/ACK packet to said client computer;

receiving, by the first BTCP module, an ACK packet from said client computer at said first BTCP module;

sending said ACK packet to said first TCP module;

storing said SYN packet, said ACK packet and said web request at said first server computer.

13. (Currently Amended) The method as described in Claim [[11]]12, wherein said ~~step e)~~steps f) and g) comprise~~comprises~~ the steps of:

sending a handoff request packet from said first BTCP module to said second BTCP module over said control channel;

including said SYN packet and said ACK packet in said handoff request packet;

changing a ~~first~~ destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module;

sending, by said second BTCP module, said SYN packet to ~~said~~ a second TCP module at said selected server computer;

receiving a second SYN/ACK packet at said second BTCP module;

parsing said second initial TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session;

changing a ~~second~~ destination IP address of said ACK packet to said second IP address, at said second BTCP module;

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session;

sending said ACK packet that is updated from said second BTCP module to said second TCP module; and

sending a handoff acknowledgment message from said second BTCP module to said first BTCP module.

14. (Currently Amended) The method as described in Claim 13, wherein said ACK packet includes said first initial TCP state of said first server computer ~~as provided for in step f)~~.

15. (Currently Amended) The method as described in Claim 13, wherein said handoff acknowledgment message includes said ~~second initial TCP state of said second server computer, including a~~ second initial sequence number~~, for said second TCP module, that is associated with said TCP/IP communication session as provided for in step i)~~.

16. (Currently Amended) The method as described in Claim 13, comprising the further steps of:

intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module at said first server computer, said connection indication message sent by said first TCP module upon establishing said communication session; and

holding said connection indication message at said first UTCP module, wherein said first UTCP module and said first BTCP module provide a wrapper around said first TCP module.

17. (Currently Amended) The method as described in Claim 16, ~~wherein step h) comprises~~ comprising the further steps of:

sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment ~~packet~~ message to said first TCP module;

discarding said connection indication message at said first UTCP module;

receiving incoming data packets from said client at said first BTCP module;

changing ~~said~~ destination addresses of said incoming data packets to said second IP address;

updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer; and

forwarding said updated data packets to said selected server computer.

18. (Currently Amended) The method as described in Claim 11, wherein step k) comprises the steps of:

intercepting outgoing response packets from said selected server computer at said second BTCP module;

changing, by said second BTCP module, source addresses of said response packets to said first IP address;

updating, by said second BTCP module, sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer; and

sending said updated response packets to said client computer without passing the updated response packets through said first server computer.

19. (Currently Amended) The method as described in Claim 11, wherein step l) comprises the steps of:

monitoring TCP/IP control traffic for said communication session at said second BTCP module;

understanding when said communication session is closed at said ~~second~~ selected server computer;

sending a termination message to said first server computer over said control channel;

terminating a forwarding mode at said first BTCP module; and

freeing data resources associated with said communication session at said first server computer.

20. (Currently Amended) The method as described in Claim 16, comprising the further steps of:

sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer; and

sending incoming data packets, including said web request, from said client computer, received at said first BTCP module, upstream.

21. (Currently Amended) The method as described in Claim 11, wherein each of said plurality of server computers is constructed similarly including BTCP modules located downstream from respective TCP modules in the corresponding server computers, and UTCP modules located upstream from the corresponding TCP modules, wherein each pair of the BTCP modules and UTCP modules provides a wrapper around a corresponding TCP module.

22. (Cancelled)

23. (Currently Amended) The method as described in Claim [[22]]21, wherein said control channel allows for communication between all UTCP modules of the corresponding server computers.

24. (Original) The method as described in Claim 11, wherein said plurality of server computers is coupled together over a wide area network in said communication network.

25. (Original) The method as described in Claim 11, wherein said information is partitioned/partially replicated throughout each of said plurality of server computers.

26. (Currently Amended) A first server computer comprising:

an operating system;

a TCP module in the operating system;

~~an~~ a first upper TCP (UTCP) module located above [[a]]the TCP module in [[an]]the operating system of said first server computer;

a first bottom TCP (BTCP) module located below said TCP module, wherein said first UTCP, TCP, and first BTCP modules implement ~~implementing a method of~~ handing off a communication session between the first server computer and a second node, wherein the first BTCP and first UTCP modules are dynamically loadable kernel modules (DLKMs) that are dynamically loadable and unloadable in said first server computer without modifying the operating system~~a first node and second node in a cluster network that works within the kernel level of an existing TCP/IP protocol, by migrating TCP states associated with said first and second nodes~~.

27. (Currently Amended) The first server computer as described in Claim 26, wherein said ~~method comprises the steps of:~~

~~a) establishing a TCP/IP~~ communication session is established between a client computer and said server computer, ~~said first node,~~ said first server computer part of a plurality of server computers including said second node forming ~~said~~ a cluster that contains ~~network containing~~ information, said communication session established for the transfer of data contained within said information;

~~b) receiving~~ wherein said first BTCP module is configured to:

receive a web request associated with said TCP/IP communication session ~~at a first BTCP module at said server computer~~;

~~c) examining~~ examine content of said web request;

~~d) determining~~ determine which of said plurality of server computers~~,~~ ~~a selected server computer,~~ can best process said web request, based on said content, wherein the server computer determined to be able to best process said web request is the second node;

~~e) handing~~ hand off said communication session to said ~~selected server computer~~ second node from said first server computer over a persistent control channel, if said ~~selected server computer~~ second node is not said first server computer, wherein the handing off causes migration of a first TCP state of said first server computer to the second node, and migration of a second TCP state of said second node to the server computer~~; and~~

~~f) migrating a first TCP state of said server computer to said selected server computer, and sending a second TCP state of said selected server computer to said server computer over said control channel~~.

28. (Currently Amended) The first server computer as described in Claim 27, wherein said first BTCP module is configured to further ~~step a) of said method comprises the steps of~~:

~~receiving~~ receive a SYN packet from said client computer ~~at said BTCP module~~;

~~sending~~ send said SYN packet upstream to said TCP module;

~~receiving~~ receive a first SYN/ACK packet from said TCP module;

~~parsing~~ parse a first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said TCP module associated with said TCP/IP communication session;

~~sending~~ send said SYN/ACK packet to said client computer;

~~receiving~~ receive an ACK packet from said client computer ~~at said BTCP module~~;

~~sending~~ send said ACK packet to said TCP module~~;~~

~~storing said SYN, ACK at said server computer~~.

29. (Currently Amended) The first server computer as described in Claim 28, wherein said first BTCP module is configured to further ~~method comprises the steps of~~:

~~sending~~ send a handoff request packet ~~from said BTCP module~~ to a second BTCP module over said control channel, said second BTCP module located below a second TCP module in a second operating system at said second node ~~selected server computer~~;

~~including~~ include said SYN packet and said ACK packet in said handoff request packet;

~~receiving~~ receive a handoff acknowledgment message ~~at said BTCP module~~ from said second BTCP module.

30. (Cancelled)

31. (Currently Amended) The first server computer as described in Claim 29, wherein said first UTCP module is configured to ~~method comprises the further steps of~~:

~~intercepting~~ intercept a connection indication message sent from said ~~first~~ TCP module to an application layer above said ~~first~~ TCP module ~~at a first upper TCP (UTCP) module~~, said connection indication message sent by said ~~first~~ TCP module upon establishing said communication session; and

~~holding~~ hold said connection indication message ~~at said first UTCP module~~.

32. (Currently Amended) The ~~computer~~ first server system as described in Claim 31, wherein said first BTCP module is configured to further ~~method comprises the further steps of~~:

~~sending~~ send a reset packet ~~from said first BTCP module~~ upon receiving said handoff acknowledgment message ~~packet~~ to said ~~first~~ TCP module;

~~discarding said connection indication message at said first UTCP module;~~

~~receiving~~ receive incoming data packets from said client computer ~~at said first BTCP module~~;

~~changing said~~ change destination addresses of said incoming data packets to ~~said~~ a second IP address of said second node;

~~updating~~ update sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said second node~~selected server computer~~; and

~~forwarding~~ forward said updated data packets to said second node~~selected server computer~~.

33. (Currently Amended) The first server computer as described in Claim 31, wherein said first BTCP module is configured to further ~~said method comprising the further steps of~~:

~~sending~~ send notification ~~from said BTCP module~~ to said first UTCP module to release said connection indication message, if said second node~~selected server computer~~ is said first server computer;

~~sending~~ send incoming data packets, including said web request, from said client computer, received at said first BTCP module, upstream.

34. (Currently Amended) The first server computer as described in Claim 26, wherein said first BTCP module is configured to ~~said method comprising the further steps of~~:

~~receiving~~ receive a handoff request from a ~~first~~ second BTCP module located at a ~~first~~ second server computer ~~within said cluster network~~ over a persistent control channel, said ~~first~~ second server computer having established a second communication session with a second client computer, ~~said communication session established for the transfer of data contained within said server computer,~~ said handoff request including a SYN packet and an ACK packet, said SYN packet and ACK packet used for establishing said second communication session between said second client computer and said ~~first~~ second server computer, said ACK packet including a first initial TCP state including a first initial TCP sequence number of said ~~first~~ second server computer in said communication session~~, including a first initial TCP sequence number~~;

~~changing~~ change a ~~first~~ destination IP address of said SYN packet to a ~~second~~ first IP address of said first server computer~~, at said BTCP module~~;

~~sending~~ send said SYN packet to said TCP module;

~~receiving~~ receive a SYN/ACK packet ~~at said second BTCP module~~;

~~parsing~~ parse a second initial TCP state from ~~second~~ said SYN/ACK packet, including a second initial sequence number, for said TCP module~~, said second initial TCP state associated with a second TCP state for said server computer in said TCP/IP communication session~~;

~~changing~~ change a ~~second~~ destination IP address of said ACK packet to said ~~second~~ first IP address~~, at said BTCP module~~;

~~updating~~ update said ACK packet to reflect said second TCP state ~~of said selected server computer in said communication session~~;

~~sending~~ send said ACK packet that is updated to said TCP module; and

~~sending~~ send a handoff acknowledgment message to said ~~first~~ second BTCP module over said control channel.

35.-37. (Cancelled)

38. (New) The method as described in Claim 1, wherein the TCP handoff modules are dynamically loadable kernel modules, the method further comprising:

dynamically loading the TCP handoff modules in the corresponding first server computer and selected server computer without modifying the operating systems of the respective first server computer and selected server computer.

39. (New) The method as described in claim 6, wherein the first BTCP module and first UTCP module are dynamically loadable kernel modules.